Amendments to the Claims

Please amend claims to be as follows.

1. (currently amended) A method of compiling a program to be executed on a target microprocessor, the method comprising:

identifying, while compiling the ~~progam~~ program, a cycle during which a functional unit in the microprocessor would otherwise be idle;

opportunistically scheduling, during compilation of the program, a diagnostic operation for execution on the functional unit during said identified cycle; and

scheduling, during compilation of the program, a comparison of a result from executing the diagnostic operation with a corresponding predetermined result.

2. (original) The method of claim 1, further comprising:

predetermining a test pattern of diagnostic operations and corresponding predetermined results for the functional unit.

3. (original) The method of claim 1, further comprising:

causing a flag in the target microprocessor to be set when the comparison indicates an error.

4. (original) The method of claim 3, further comprising:

if the error flag is set, then halting the execution and causing a notification to a user of the error flag.

5. (original) The method of claim 1, further comprising:

setting a user-selectable level ('slider') for an aggressiveness of said opportunistic scheduling.

6. (original) The method of claim 1, wherein the functional unit comprises a floating point unit.

7. (original) The method of claim 1, wherein the functional unit comprises an arithmetic logic unit.

8. (original) The method of claim 1, wherein the functional unit comprises one of multiple functional units of a same type within the target microprocessor.

9. (original) The method of claim 1, wherein the method is performed by a scheduler in a code generator of a program compiler.

10. (original) The method of claim 9, wherein the program compiler comprises a native compiler for the target microprocessor.

11. (original) The method of claim 9, wherein the program compiler comprises a cross compiler run on a different microprocessor.

12. (canceled)

13. (previously presented) A program compiler stored on a computer-readable medium for use with a target microprocessor, the compiler comprising a code generator

including a scheduler that identifies a cycle during which a functional unit would otherwise be idle, opportunistically schedules a diagnostic operation to be executed on the functional unit during that cycle, and schedules a comparison of a result from executing the diagnostic operation with a corresponding predetermined result, wherein the functional unit comprises one of multiple functional units of a same type within the target microprocessor.

14. (previously presented) A program compiler stored on a computer-readable medium for use with a target microprocessor, the compiler comprising a code generator including a scheduler that identifies a cycle during which a functional unit would otherwise be idle, opportunistically schedules a diagnostic operation to be executed on the functional unit during that cycle, and schedules a comparison of a result from executing the diagnostic operation with a corresponding predetermined result, wherein the scheduler selects the diagnostic operation from a test pattern of diagnostic operations.

15. (canceled)